

## TRABAJO PRACTICO INTEGRADOR PROGRAMACION I

**Modalidad de Entrega:** Principalmente lo que se espera es un documento entregable, en formato WORD o PDF y el programa en funcionamiento.

En relación al documento, debe tener una portada con logo de la UCEMA y los datos de cada uno de los alumnos que conforman el grupo.

El trabajo debe tener un índice de contenidos tal como se describe a continuación:

**Introducción:** Aquí se debe explicar una síntesis de lo que se realiza en el proyecto. No hay que copiar y pegar el enunciado, sino que hay que sintetizar el problema a resolver, pudiendo adjuntar alguna imagen o diagrama que ayude a dicha explicación.

**Diseño conceptual:** Aquí lo que se espera es que el grupo pueda explicar en palabras, cómo hará para resolver el problema desde el punto de vista del diseño del programa. Se deberán adjuntar diagramas de flujo, diagramas de bloques o lo que el grupo considere mas pertinente. El objetivo es pensar, explicar y dejar preparado el diseño de la solución al problema presentado.

**Descripción de las funciones:** Aquí se deberán explicar cómo funciona cada una de las funciones del programa. No sólo desde el punto de vista de los parámetros que recibe y valores que devuelve, sino que además, se logre entender la relación y la importancia de cada una de las funciones que se programarán.

**Pruebas de funcionalidad:** Aquí se debe armar una lista de no menos de 5 pruebas elementales que se deberán hacer sobre el programa para verificar el correcto funcionamiento del mismo. La estructura de la lista debe seguir los siguiente campos (se puede hacer en Excel).

ID de la prueba	Descripción	Función a probar	Requisito para la prueba	Resultado esperado	Resultado obtenido	Paso (SI/NO)
1	Carga de producto	Cargar_Producto()	que el programa esté en ejecución	Que se cargue un producto en un archivo	producto cargado (ver imagen1)	SI

**Manual de usuario:** Aquí se deben adjuntar las capturas de pantalla y la explicación de cómo operar el programa. Hacer de cuenta que es un manual explicativo del uso del programa.

**Código Fuente comentado:** Adjuntar todo el código del programa, comentado, y además, con claras separaciones entre las diferentes funciones. Del mismo.

### Comentarios sobre el problema a resolver.

Para resolver el problema que se presenta, no existe una única forma. Habrá detalles y decisiones que, en caso de no estar explicitadas en el enunciado, uds. Como grupo, deberán decidir las. El grupo podrá tomar todas las decisiones que considere pertinentes, desde cuestiones básicas como tipos de datos a utilizar, hasta que gráficos incorporar y la cantidad de Structs a utilizar. Aquí esperamos que cada grupo resuelva de la manera mas creativa, utilizando todos los recursos que se vieron a lo largo de la materia, y sobre todo considerando la persistencia de datos en archivos, tanto binarios, como de texto.

# CASOS

## Descripción del CASO Nro 1: Empresa Metalúrgica

Una empresa metalúrgica le encarga a usted la construcción de un sistema de ventas, el cual, por una cuestión de funcionamiento, se le solicita que la información sea persistida en archivos.

El Sistema de ventas contiene un módulo de clientes, de los cuales se tiene

*Razon\_social*

*Direccion*

*Rubro*

*Cant\_Días\_sin\_Comprar* (por defecto, está en 9999. Cada vez que se compra algo ,se pone en cero y se incrementa 1 a 1. Cada vez que se cierra el programa, se termina un día).

*Cant\_facturacion* (Es la acumulación de la facturación x los productos que compró)

*Ciente\_VIP*

El campo Dirección a su vez, está compuesta por

*Calle*

*Altura*

*Piso*

Se pide que construya un programa que sea capaz de *gestionar a los clientes* de esta empresa metalúrgica.

La gestión implica: la carga de nuevos clientes, la modificación de los actuales y su consecuente actualización en archivos que es donde persiste la información.

El programa deberá tener un menú de opciones no sólo para gestionar, sino que además cada cliente a su vez puede hacer una compra de uno o mas productos.

Dichos productos tienen las siguientes características:

*Nombre* (Ej chapa doble)

*Descripción* (Ej chapa galvanizada doble de 1.2x2.5mts)

*Precio*

*Cant\_en\_stock*

Cada vez que un cliente compra, Ingresa el nombre del producto a comprar (el producto debe existir en un archivo). El programa genera el ticket de la compra, lo muestra en pantalla, y además, cada ticket lo exporta a un archivo de texto, de manera que se pueda tener el historial de toda la facturación de la empresa.

Este programa deberá considerar que un cliente VIP es aquel que sobre el ticket de compra recibe un 10% de descuento. Para que un cliente sea VIP tiene que superar los \$800.000 pesos de facturación emitida.

El programa deberá poder imprimir en pantalla aquellos clientes que no tengan compras realizadas en los últimos X cantidad de días, siendo X un numero que ingresa el usuario por teclado.

El programa debe generar reportes de productos mas vendidos, clientes VIP y de facturación emitida Sumará puntaje extra el hecho de reportar facturación día a día considerando que un día termina cuando el programa se cierra.

Todo debe ser programado utilizando memoria dinámica, listas, colas, o pilas y todo debe persistir en archivos de texto y/o archivos binarios.

## Descripción del CASO Nro 2: Problema del Banco

UD trabaja en un banco, y se le pide que realice un programa que permita ingresar los siguientes datos de los clientes:

*ID de cliente*  
*Nombre\_Cliente*  
*Numero de Cuenta*  
*Ciudad del cliente*  
*Balance de la cuenta*

Esos datos serán guardados en un archivo llamado “clientes.dat”.

El programa además de cargar nuevos clientes, debe permitir leer del archivo “clientes.dat” los datos que tiene cargados, y mediante un menú interactivo, se deben listar los datos cargados en el archivo “clientes.dat”, filtrando por balance (Cero, Negativo, Positivo).

Adicionalmente se debe permitir la modificación de los datos de cada cliente, sobre todo la modificación de balance, no así el ID de cliente y el numero de cuenta.

A su vez, el programa debe generar otro archivo binario, llamado **reporte.dat** que permita, utilizando una lista enlazada, tomar la información del archivo clientes.dat y que permita mostrar los clientes ordenados por Id de cliente, con el balance de cada uno de ellos.

En un tercer archivo, que se generará cuando el usuario seleccione la opción “reporte”, se deberá almacenar en un archivo de texto (**reporte.txt**) todos los clientes que tienen saldo negativo, ordenados por ID de cliente. Este programa, para que funcione correctamente, deberá disponer de un menú de opciones que ud. deberá diseñar sobre la base de lo explicado anteriormente.

Por ultimo, se debe llevar un log de transacciones, indicando el nombre de usuario (nombre y apellido) del operador del bando que operó el sistema.

Sumará puntaje extra si se logra eliminar un registro de un cliente y que el mismo quede actualizado en los archivos.

Los tipos de datos, así como cualquier otra información que no esté asociada en este detalle, y que sea necesaria, deberá definirla ud. mismo con su respectiva justificación.

### Descripción del CASO Nro 3: La empresa Industrial

Una empresa industrial fabrica y vende 5 productos de acero que se utilizan para cualquier tipo de industria, sobre todo, relacionada a la construcción.

Si bien, en la actualidad se fabrican y venden 5 tipos de productos, no se descarta que a futuro se puedan vender mas. Los productos residen en un archivo binario PRODUCTOS.BIN

La estructura productos

*ID\_producto*

*Nombre*

*Descripcion*

*Costo de fabricación*

*Precio de venta*

*Cant\_en\_stock*

Los productos se fabrican a medida que se piden. El primer producto en pedirse es el primero en fabricarse. El programa debe manejar la gestión de pedidos de productos, generar ticket de facturación y persistir toda la información en archivos binarios y de texto, según se crea conveniente.

Para poder hacer pedidos, los clientes previamente tienen que existir, si no existen se deben cargar en el archivo CLIENTE.BIN

Los datos que se deben almacenar son:

*Id Cliente*

*Nombre Cliente (razon social)*

*Ubicación\_Cliente*

Ubicación\_Cliente:

*Id ubicación*

*Calle*

*nro*

*CP*

*Ciudad*

*País*

El programa debe reportar a todos los clientes registrados, y los debe mostrar en pantalla ordenados por ID de clientes. De menor a mayor.

El programa debe ser capaz de registrar todas las operaciones que se realicen mientras el programa esté en ejecución (Log de actividades)

El programa deberá reportar productos ordenados por ID de producto.

El precio de venta de los productos debe ser calculado automáticamente a partir del costo de fabricación. Dicho importe es un 50% mas....y se debe considerar que ese valor debe ser fácil de reemplazar, en función de las necesidades del negocio.

En un archivo llamado reporte.txt se debe permitir generar, cuando el usuario del programa lo disponga (seleccionando la opción correspondiente) un reporte de la cantidad de dinero vendida por día, indicando cual fué el cliente que más compró (en términos de dinero y no en términos de cantidad de productos), e indicar de que país y ciudad es el cliente que más compró.

#### Descripción del CASO Nro 4: La veterinaria

Un médico veterinario, tiene un consultorio y para poder trabajar en su día a día, necesita de un programa que permita registrar a las mascotas que atiende.

Las Mascotas:

ID mascota  
Nombre de pila  
Raza  
Teléfono de contacto  
Ficha\_médica

Ficha medica es otra estructura anidada, donde únicamente se vuelcan anotaciones de texto libre para que el veterinario pueda leer todas las anotaciones que hizo sobre una mascota en particular

Se debe considerar que el veterinario también puede vender productos. Para eso deberá contar con un listado de productos que deben persistir en un archivo llamado PRODUCTOS.BIN

El programa debe gestionar la venta de los productos y emitir un ticket, en cuyo caso la información de la venta diaria debe persistir en un archivo de texto llamado FACTURACION.txt

El programa además, debe poder reportar mascotas ordenadas por ID y debe permitir buscar mascotas por nombre o raza

Se considerará un plus, la implementación de una función que permita ayudar al veterinario en su consultorio (ej. Mediante algun reporte nuevo o una función de utilidad para el médico).

Notas adicionales:

- La información de productos y de mascotas debe poder modificarse.
- Las mascotas se atienden por orden de llegada. Y cada vez que una mascota es atendida debe quedar un log que reporte lo que el veterinario atendió día a día, por orden de llegada.x

### Descripción del CASO Nro 5: Estacionamiento de vehículos

Un estacionamiento muy pequeño, tiene capacidad para alojar a 12 vehículos. El estacionamiento se cobra por minuto. Cada minuto cuesta \$5. Dicho importe tiene que poder ser modificado si fuera necesario.

Se necesita hacer un programa que permita gestionar el estacionamiento, y que esa gestión incluya al menos:

- 1) Conocer en todo momento, si hay disponibilidad o no para que un auto ingrese
- 2) Cobrar cuando un vehículo se retira, en función del tiempo transcurrido.
- 3) Reportar en todo momento la cantidad de autos estacionados, y la cantidad de lugares disponibles
- 4) Reportar la facturación de un día y exportarla a un archivo
- 5) Registrar todos los vehículos que ingresan por nro de patente.
- 6) Reportar todos los autos que ingresaron en un día ordenados por orden de llegada y cuando se facturó a cada uno.
- 7) Gestionar la fila de espera. En caso de que estén ocupados todos los lugares, los vehículos pueden esperar (hacer fila) para ingresar.
- 8) Los autos se recomienda gestionarlos con listas
- 9) El programa debe loguear todas las acciones que se realizan y persistir la información en un archivo de texto

Todas aquellas definiciones respecto del proceso de como llega un auto, como se retira, como se cobra, como se reporta, etc., que no esté explícito en la consigna, la debe determinar el grupo en función de lo mas conveniente desde el punto de vista de la realización del programa y de la gestión del dueño del estacionamiento.

Las definiciones de estructuras de datos, miembros de structs y demás información la debe definir el grupo en función de las definiciones que se tomen.

### Descripción del CASO Nro 6: Gestión de Entradas al teatro

Se trata de hacer un programa que permita gestionar la entrada a un evento de teatro.

Este teatro es muy particular, porque registra a los clientes antes de vender la entrada al espectáculo. Los datos que registra son: el dni, el nombre y apellido, además de identificar al mismo como VIP o NO VIP.

Si un cliente es VIP, recibe un 20% de descuento al momento de pagar la entrada.

El teatro tiene una capacidad máxima de 200 personas, por lo que el sistema no podrá vender mas de 200 entradas.

Cada vez que una entrada se vende, a un precio fijo, (Este precio fijo debe ser fácil de modificar), se asigna el número de asiento, el cual es un numero entero, consecutivo del 1 al 200.

Piense a la cola de clientes como una cola dinámica, y a los asientos del teatro como una estructura pila. Cuando la función del teatro finaliza, los clientes se van en el siguiente orden: desde el ultimo que ingresó al primero que lo hizo (desapilar).

El programa debe ir mostrando progresivamente como se van los clientes del teatro, mostrando en pantalla (con un delay de 2 segundos) los datos de los clientes que se van. EJ:

200 .....2s .....199.....2s .....198 .....2s etc.

Piense en un algoritmo que permita establecer cuando el cajero está ocupado. Cuando eso sucede, si justo se quiere llegar otro cliente debe esperar a que pueda ser atendido, es decir, se debe encolar.

El menú de opciones del programa debe permitir entre todas las opciones mencionadas, almacenar en archivos los siguientes datos:

Cantidad de entradas vendidas

Cantidad total de dinero recaudado

Cantidad total de clientes VIP que asistieron a la función

Además en cualquier momento el cajero puede ver cuantas entradas disponibles quedan.



### Descripción del CASO Nro 7: Gestión de pedidos comidas rápidas

Se trata de un sistema para gestionar la venta de un local de comidas rápidas, que solo vende dos productos (Salchichas y/o hamburguesas).

Los productos tienen la siguiente estructura

ID\_Producto: Es un numero entero y consecutivo que comienza con el 101.

Nombre\_Producto:

Precio\_Producto

Cantidad:

El pedido que realiza un cliente, se realiza de un único producto por vez (uno a la vez).

Los pedidos se ingresan a una cola de espera, mientras son preparados.

En el menú principal del programa se debe mostrar en todo momento, la cantidad de elementos que están siendo preparados.

Cuando el producto está listo para ser retirado, el operador del sistema, indica que el producto está listo, y el mismo sale de la cola, para incorporarse como información a un archivo donde se registran todos los productos que fueron vendidos y entregados. Indicando para esto, los datos de los producto, con las cantidades y el precio.

El sistema debe tener la posibilidad de emitir reportes de cuantos productos están en preparación. Cuantos productos fueron entregados hasta el momento y al finalizar (cuando el sistema termina o cuando se sale del mismo) se debe crear un archivo log.txt donde se indiquen dos renglones:

- 1) Cantidad de productos vendidos y
- 2) cantidad de dinero facturado.

NOTA: Tendrá un puntaje adicional el hecho de hacer este programa, y que dentro del pedido se puedan incorporar varios productos para el mismo cliente.